

In the Claims:

Please amend the claims as indicated below:

1. (Withdrawn) A Web Service system, comprising:

a service provider configured to provide one or more services on the Web Service system;

one or more service requesters configured to access the one or more services from the service provider via a network;

wherein the Web Service system is configured according to a vendor-independent architecture framework for designing Web Services comprising a plurality of heterogeneous components in accordance with a structured methodology and one or more design patterns.

2. (Withdrawn) The Web Service system as recited in claim 1, wherein the architecture framework is configured to incorporate Quality of Services including reliability, scalability, and availability on the Web Service system.

3. (Withdrawn) The Web Service system as recited in claim 1, further comprising a service broker configured to interact with the service provider and service requester to negotiate and provide the services of the service provider to the service requester.

4. (Withdrawn) The Web Service system as recited in claim 3, wherein the service provider is configured to serve as the service broker.

5. (Withdrawn) The Web Service system as recited in claim 1, further comprising a service registry, wherein the service provider is further configured to

register and publish the services in the service registry, and wherein the service requester is further configured to discover the service provider through the service registry.

6. (Withdrawn) The Web Service system as recited in claim 1, wherein the Web Service system is a Business-to-Consumer Web Service, wherein the service provider is a business service provider, and wherein the service requester is an end user.

7. (Withdrawn) The Web Service system as recited in claim 1, wherein the Web Service system is a Business-to-Business Web Service, wherein the service provider is a business service provider, and wherein the service requester is a server.

8. (Withdrawn) The Web Service system as recited in claim 1, wherein the Web Service system comprises a plurality of layers, wherein the plurality of layers comprises two or more of:

- a network layer configured to serve as an underlying network for services;
- a transport layer for delivering messages between components of the Web Service system;
- a service description language layer configured to describe service type and functionality of the services of the service provider;
- a transaction routing layer configured to route messages on the transport layer;
- a service discovery layer configured to search for and locate services;
- a service negotiation layer configured to negotiate exchanges between the service requesters and the service provider;
- a management layer configured for provisioning of the services and for monitoring and administration of the services;
- a Quality of Service layer configured to provide reliability, scalability, and availability on the Web Service system;
- a security layer configured to provide authentication, entitlement, and non-repudiation security on the transport layer; and
- an Open Standards layer.

9. (Withdrawn) The Web Service system as recited in claim 8, wherein the Open Standards layer is XML.

10. (Withdrawn) The Web Service system as recited in claim 8, wherein the network layer is the public Internet over TCP/IP.

11. (Withdrawn) The Web Service system as recited in claim 8, wherein the transport layer is one of HTTP, SMTP, and SOAP over HTTP.

12. (Withdrawn) The Web Service system as recited in claim 1, wherein the design patterns include one or more of:

- one or more scalability design patterns;
- one or more reliability design patterns
- one or more manageability design patterns;
- one or more availability design patterns; and
- one or more security design patterns.

13. (Withdrawn) The Web Service system as recited in claim 1, wherein the design patterns include one or more of:

- one or more Quality of Services design patterns;
- one or more Integration design patterns; and
- one or more Security design patterns.

14. (Withdrawn) The Web Service system as recited in claim 13, wherein the Quality of Services design patterns include one or more of:

- a SOAP Cache Design Pattern;
- a Java Message Service (JMS) Bridge Design Pattern;
- a Multiple Servlet Engines Design Pattern;
- an HTTP Load Balancer Design Pattern;
- a State Management Design Pattern;
- a SOAP Logger Design Pattern;

- a High Availability of Service Registry Design Pattern;
- a UDDI Deployment Design Pattern;
- a Publish, Unpublish, and Discover Web Services Design Pattern;
- a Version Management of Deployment and Service Registry Design Pattern; and
- a Registry Content Management Design Pattern.

15. (Withdrawn) The Web Service system as recited in claim 13, wherein the Integration design patterns include one or more of:

- an Application-to-Application Design Pattern;
- a Standard Build Design Pattern;
- a Hub-Spoke Replication Design Pattern;
- a Federated Replication Design Pattern;
- a Multi-Step Application Integration Design Pattern;
- a Data Exchange Design Pattern;
- a Closed Process Integration Design Pattern;
- an Open Process Integration Design Pattern;
- a Service Consolidation–Broker Integration design pattern; and
- a Reverse Auction–Broker Integration design pattern.

16. (Withdrawn) The Web Service system as recited in claim 13, wherein the Security design patterns include one or more of:

- a Single Sign-on Design Pattern; and
- a Messaging Transport Design Pattern.

17. (Withdrawn) The Web Service system as recited in claim 1, wherein the Web Service system comprises two or more of:

- a client tier;
- a presentation tier;
- a business tier;
- an integration tier; and
- a resource tier.

18. (Original) A system for generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components, comprising:

means for generating one or more Use Cases for the Web Service in accordance with one or more design patterns;

means for generating a high-level architecture for the Web Service and in accordance with the one or more design patterns, wherein the high-level architecture identifies two or more entities of the Web Service and the relationships and interactions among the entities; and

means for generating a logical architecture for the Web Service according to the use case scenarios and in accordance with the one or more design patterns, wherein the logical architecture identifies two or more logical components of the Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers.

19. (Original) The system as recited in claim 18, wherein the Web Service architecture incorporates Quality of Services including reliability, scalability, and availability on the Web Service system.

20. (Original) A method, comprising:

generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components in accordance with one or more design patterns, wherein said generating a vendor-independent Web Services architecture comprises:

generating one or more Use Cases for the Web Service;

generating a high-level architecture for the Web Service, wherein the high-level architecture identifies two or more entities of the Web Service and the relationships and interactions among the entities;

generating a logical architecture for the Web Service according to the use case scenarios, wherein the logical architecture identifies two or more logical components of the Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers; and

implementing the Web Service according to the Web Service architecture.

21. (Original) The method as recited in claim 20, wherein said generating a high-level architecture for the Web Service comprises identifying one or more Open Standards protocols for use in said interactions among the entities.

22. (Original) The method as recited in claim 20, wherein the Web Service architecture incorporates Quality of Services including reliability, scalability, and availability on the Web Service.

23. (Original) The method as recited in claim 20, wherein the entities comprise:

a service provider configured to provide one or more services on the Web Service; and

one or more service requesters configured to access the one or more services from the service provider via a network.

24. (Original) The method as recited in claim 23, wherein the entities further comprise a service broker configured to interact with the service provider and service

requester to negotiate and provide the services of the service provider to the service requester.

25. (Original) The method as recited in claim 23, wherein the entities further comprise a service registry, wherein the service provider is further configured to register and publish the services in the service registry, and wherein the service requester is further configured to discover the service provider through the service registry.

26. (Original) The method as recited in claim 23, wherein the Web Service is a Business-to-Consumer Web Service, wherein the service provider is a business service provider, and wherein the service requester is an end user.

27. (Original) The method as recited in claim 23, wherein the Web Service is a Business-to-Business Web Service, wherein the service provider is a business service provider, and wherein the service requester is a server.

28. (Original) The method as recited in claim 23, wherein the layers of the logical architecture comprise two or more of:

- a network layer configured to serve as an underlying network for services;
- a transport layer for delivering messages between components of the Web Service;
- a service description language layer configured to describe service type and functionality of the services of the service provider;
- a transaction routing layer configured to route messages on the transport layer;
- a service discovery layer configured to search for and locate services;
- a service negotiation layer configured to negotiate exchanges between the service requesters and the service provider;
- a management layer configured for provisioning of the services and for monitoring and administration of the services;
- a Quality of Service layer configured to provide reliability, scalability, and availability on the Web Service;

a security layer configured to provide authentication, entitlement, and non-repudiation security on the transport layer; and
an Open Standards layer.

29. (Original) The method as recited in claim 28, wherein the Open Standards layer is XML.

30. (Original) The method as recited in claim 28, wherein the network layer is the public Internet over TCP/IP.

31. (Original) The method as recited in claim 28, wherein the transport layer is one of HTTP, SMTP, and SOAP over HTTP.

32. (Original) The method as recited in claim 20, wherein the design patterns include one or more of:

- one or more scalability design patterns;
- one or more reliability design patterns
- one or more manageability design patterns;
- one or more availability design patterns; and
- one or more security design patterns.

33. (Original) The method as recited in claim 20, wherein the design patterns include one or more of:

- one or more Quality of Services design patterns;
- one or more Integration design patterns; and
- one or more Security design patterns.

34. (Original) The method as recited in claim 33, wherein the Quality of Services design patterns include one or more of:

- a SOAP Cache Design Pattern;
- a Java Message Service (JMS) Bridge Design Pattern;

- a Multiple Servlet Engines Design Pattern;
- an HTTP Load Balancer Design Pattern;
- a State Management Design Pattern;
- a SOAP Logger Design Pattern;
- a High Availability of Service Registry Design Pattern;
- a UDDI Deployment Design Pattern;
- a Publish, Unpublish, and Discover Web Services Design Pattern;
- a Version Management of Deployment and Service Registry Design Pattern; and
- a Registry Content Management Design Pattern.

35. (Original) The method as recited in claim 33, wherein the Integration design patterns include one or more of:

- an Application-to-Application Design Pattern;
- a Standard Build Design Pattern;
- a Hub-Spoke Replication Design Pattern;
- a Federated Replication Design Pattern;
- a Multi-Step Application Integration Design Pattern;
- a Data Exchange Design Pattern;
- a Closed Process Integration Design Pattern;
- an Open Process Integration Design Pattern;
- a Service Consolidation–Broker Integration design pattern; and
- a Reverse Auction–Broker Integration design pattern.

36. (Original) The method as recited in claim 33, wherein the Security design patterns include one or more of:

- a Single Sign-on Design Pattern; and
- a Messaging Transport Design Pattern.

37. (Original) The method as recited in claim 20, wherein the logical architecture further comprises a plurality of tiers, wherein the tiers comprise two or more of:

- a client tier;

a presentation tier;
a business tier;
an integration tier; and
a resource tier.

38. (Currently amended) A computer-accessible storage medium storing ~~comprising~~ program instructions, wherein the program instructions are ~~configured~~ computer-executable to implement:

generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components in accordance with one or more design patterns, wherein said generating a vendor-independent Web Services architecture comprises:

generating one or more Use Cases for the Web Service;

generating a high-level architecture for the Web Service, wherein the high-level architecture identifies two or more entities of the Web Service and the relationships and interactions among the entities;

generating a logical architecture for the Web Service according to the use case scenarios, wherein the logical architecture identifies two or more logical components of the Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers; and

implementing the Web Service according to the Web Service architecture.

39. (Currently amended) The computer-accessible storage medium as recited in claim 38, wherein, in said generating a high-level architecture for the Web Service, the program instructions are further ~~configured~~ computer-executable to implement

identifying one or more Open Standards protocols for use in said interactions among the entities.

40. (Currently amended) The computer-accessible storage medium as recited in claim 38, wherein the Web Service architecture incorporates Quality of Services including reliability, scalability, and availability on the Web Service.

41. (Currently amended) The computer-accessible storage medium as recited in claim 38, wherein the entities comprise:

a service provider configured to provide one or more services on the Web Service; and

one or more service requesters configured to access the one or more services from the service provider via a network.

42. (Currently amended) The computer-accessible storage medium as recited in claim 41, wherein the entities further comprise a service broker configured to interact with the service provider and service requester to negotiate and provide the services of the service provider to the service requester.

43. (Currently amended) The computer-accessible storage medium as recited in claim 41, wherein the entities further comprise a service registry, wherein the service provider is further configured to register and publish the services in the service registry, and wherein the service requester is further configured to discover the service provider through the service registry.

44. (Currently amended) The computer-accessible storage medium as recited in claim 41, wherein the Web Service is a Business-to-Consumer Web Service, wherein the service provider is a business service provider, and wherein the service requester is an end user.

45. (Currently amended) The computer-accessible storage medium as recited in claim 41, wherein the Web Service is a Business-to-Business Web Service, wherein the service provider is a business service provider, and wherein the service requester is a server.

46. (Currently amended) The computer-accessible storage medium as recited in claim 41, wherein the layers of the logical architecture comprise two or more of:

- a network layer configured to serve as an underlying network for services;
- a transport layer for delivering messages between components of the Web Service;
- a service description language layer configured to describe service type and functionality of the services of the service provider;
- a transaction routing layer configured to route messages on the transport layer;
- a service discovery layer configured to search for and locate services;
- a service negotiation layer configured to negotiate exchanges between the service requesters and the service provider;
- a management layer configured for provisioning of the services and for monitoring and administration of the services;
- a Quality of Service layer configured to provide reliability, scalability, and availability on the Web Service;
- a security layer configured to provide authentication, entitlement, and non-repudiation security on the transport layer; and
- an Open Standards layer.

47. (Currently amended) The computer-accessible storage medium as recited in claim 46, wherein the Open Standards layer is XML.

48. (Currently amended) The computer-accessible storage medium as recited in claim 46, wherein the network layer is the public Internet over TCP/IP.

49. (Currently amended) The computer-accessible storage medium as recited in claim 46, wherein the transport layer is one of HTTP, SMTP, and SOAP over HTTP.

50. (Currently amended) The computer-accessible storage medium as recited in claim 38, wherein the design patterns include one or more of:

- one or more scalability design patterns;
- one or more reliability design patterns
- one or more manageability design patterns;
- one or more availability design patterns; and
- one or more security design patterns.

51. (Currently amended) The computer-accessible storage medium as recited in claim 38, wherein the design patterns include one or more of:

- one or more Quality of Services design patterns;
- one or more Integration design patterns; and
- one or more Security design patterns.

52. (Currently amended) The computer-accessible storage medium as recited in claim 51, wherein the Quality of Services design patterns include one or more of:

- a SOAP Cache Design Pattern;
- a Java Message Service (JMS) Bridge Design Pattern;
- a Multiple Servlet Engines Design Pattern;
- an HTTP Load Balancer Design Pattern;
- a State Management Design Pattern;
- a SOAP Logger Design Pattern;
- a High Availability of Service Registry Design Pattern;
- a UDDI Deployment Design Pattern;
- a Publish, Unpublish, and Discover Web Services Design Pattern;
- a Version Management of Deployment and Service Registry Design Pattern; and
- a Registry Content Management Design Pattern.

53. (Currently amended) The computer-accessible storage medium as recited in claim 51, wherein the Integration design patterns include one or more of:

- an Application-to-Application Design Pattern;
- a Standard Build Design Pattern;
- a Hub-Spoke Replication Design Pattern;
- a Federated Replication Design Pattern;
- a Multi-Step Application Integration Design Pattern;
- a Data Exchange Design Pattern;
- a Closed Process Integration Design Pattern;
- an Open Process Integration Design Pattern;
- a Service Consolidation–Broker Integration design pattern; and
- a Reverse Auction–Broker Integration design pattern.

54. (Currently amended) The computer-accessible storage medium as recited in claim 51, wherein the Security design patterns include one or more of:

- a Single Sign-on Design Pattern; and
- a Messaging Transport Design Pattern.

55. (Currently amended) The computer-accessible storage medium as recited in claim 38, wherein the logical architecture further comprises a plurality of tiers, wherein the tiers comprise two or more of:

- a client tier;
- a presentation tier;
- a business tier;
- an integration tier; and
- a resource tier.

56. (Withdrawn) A method for designing and implementing a vendor-independent Web Service architecture, comprising:

identifying one or more logical components of the Web Service architecture according to one or more Use Case requirements for a Web Service;

translating the one or more use case requirements and one or more technical constraints to determine a plurality of heterogeneous Web Service components;

categorizing the Web Service components according to a Web Service architecture framework;

organizing the Web Service components according to two or more tiers and two or more layers of the Web Service architecture;

modifying one or more software components according to one or more architecture principles for each of the one or more tiers and the one or more layers;

applying one or more Web Services design patterns to the Web Service architecture where appropriate;

implementing a Web Service according to the Web Service architecture; and

assessing the quality of services of the Web Service after said implementing.

57. (Withdrawn) The method as recited in claim 56, wherein said categorizing the Web Service components according to a Web Service architecture framework comprises categorizing the Web Service components into one or more of service delivery, service management, identity/policy and services Web Service components.

58. (Withdrawn) The method as recited in claim 56, further comprising generating the one or more Use Cases for the Web Service.

59. (Withdrawn) The method as recited in claim 56, wherein the Web Service architecture incorporates Quality of Services including reliability, scalability, and availability on the Web Service.

60. (Withdrawn) The method as recited in claim 56, wherein the Web Service comprises:

- a service provider configured to provide one or more services on the Web Service; and
- one or more service requesters configured to access the one or more services from the service provider via a network; and
- a service broker configured to interact with the service provider and service requester to negotiate and provide the services of the service provider to the service requester.

61. (Withdrawn) The method as recited in claim 60, wherein the Web Service further comprises a service registry, wherein the service provider is further configured to register and publish the services in the service registry, and wherein the service requester is further configured to discover the service provider through the service registry.

62. (Withdrawn) The method as recited in claim 60, wherein the Web Service is a Business-to-Consumer Web Service, wherein the service provider is a business service provider, and wherein the service requester is an end user.

63. (Withdrawn) The method as recited in claim 60, wherein the Web Service is a Business-to-Business Web Service, wherein the service provider is a business service provider, and wherein the service requester is a server.

64. (Withdrawn) The method as recited in claim 56, wherein the layers comprise two or more of:

- a network layer;
- a transport layer;
- a service description language layer;
- a transaction routing layer;
- a service discovery layer;
- a service negotiation layer;
- a management layer;
- a Quality of Service layer;
- a security layer; and
- an Open Standards layer.

65. (Withdrawn) The method as recited in claim 56, wherein the tiers comprise two or more of:

- a client tier;
- a presentation tier;
- a business tier;
- an integration tier; and
- a resource tier.

66. (Withdrawn) The method as recited in claim 56, wherein the design patterns include one or more of:

- one or more Quality of Services design patterns;
- one or more Integration design patterns; and
- one or more Security design patterns.

67. (Withdrawn) The method as recited in claim 66, wherein the Quality of Services design patterns include one or more of:

- a SOAP Cache Design Pattern;
- a Java Message Service (JMS) Bridge Design Pattern;

- a Multiple Servlet Engines Design Pattern;
- an HTTP Load Balancer Design Pattern;
- a State Management Design Pattern;
- a SOAP Logger Design Pattern;
- a High Availability of Service Registry Design Pattern;
- a UDDI Deployment Design Pattern;
- a Publish, Unpublish, and Discover Web Services Design Pattern;
- a Version Management of Deployment and Service Registry Design Pattern; and
- a Registry Content Management Design Pattern.

68. (Withdrawn) The method as recited in claim 66, wherein the Integration design patterns include one or more of:

- an Application-to-Application Design Pattern;
- a Standard Build Design Pattern;
- a Hub-Spoke Replication Design Pattern;
- a Federated Replication Design Pattern;
- a Multi-Step Application Integration Design Pattern;
- a Data Exchange Design Pattern;
- a Closed Process Integration Design Pattern;
- an Open Process Integration Design Pattern;
- a Service Consolidation–Broker Integration design pattern; and
- a Reverse Auction–Broker Integration design pattern.

69. (Withdrawn) The method as recited in claim 66, wherein the Security design patterns include one or more of:

- a Single Sign-on Design Pattern; and
- a Messaging Transport Design Pattern.

70. (Withdrawn) A method for designing and implementing a vendor-independent Web Service architecture, comprising:

identifying and building one or more security components according to one or more Use Case requirements for a Web Service;

identifying one or more Web Service objects of the Web Service architecture to be protected;

defining an object relationship for security protection in the Web Service architecture;

identifying one or more associated trust domains, security policy and strategy, and one or more threat profiles for the Web Service architecture;

determining one or more protection schemes and measures for the Web Services objects;

applying one or more Web Services design patterns to the Web Service architecture where appropriate;

implementing a Web Service according to the Web Service architecture; and

assessing security levels of the Web Service according to two or more tiers of the Web Service architecture after said implementing.

71. (Withdrawn) The method as recited in claim 70, further comprising applying one or more Web Services security tools to implement the protection schemes.

72. (Withdrawn) The method as recited in claim 70, further comprising generating the one or more Use Cases for the Web Service.

73. (Withdrawn) The method as recited in claim 70, wherein the Web Service comprises:

a service provider configured to provide one or more services on the Web Service; and

one or more service requesters configured to access the one or more services from the service provider via a network; and

a service broker configured to interact with the service provider and service requester to negotiate and provide the services of the service provider to the service requester.

74. (Withdrawn) The method as recited in claim 73, wherein the Web Service further comprises a service registry, wherein the service provider is further configured to register and publish the services in the service registry, and wherein the service requester is further configured to discover the service provider through the service registry.

75. (Withdrawn) The method as recited in claim 73, wherein the Web Service is a Business-to-Consumer Web Service, wherein the service provider is a business service provider, and wherein the service requester is an end user.

76. (Withdrawn) The method as recited in claim 73, wherein the Web Service is a Business-to-Business Web Service, wherein the service provider is a business service provider, and wherein the service requester is a server.

77. (Withdrawn) The method as recited in claim 70, wherein the Web Service architecture comprises a plurality of layers, where the layers comprise two or more of:

a network layer;

a transport layer;

a service description language layer;

- a transaction routing layer;
- a service discovery layer;
- a service negotiation layer;
- a management layer;
- a Quality of Service layer;
- a security layer; and
- an Open Standards layer.

78. (Withdrawn) The method as recited in claim 70, wherein the tiers comprise two or more of:

- a client tier;
- a presentation tier;
- a business tier;
- an integration tier; and
- a resource tier.

79. (Withdrawn) The method as recited in claim 70, wherein the design patterns include one or more of:

- one or more Quality of Services design patterns;
- one or more Integration design patterns; and
- one or more Security design patterns.

80. (Withdrawn) The method as recited in claim 70, wherein the design patterns include one or more of:

- a Single Sign-on Design Pattern; and
- a Messaging Transport Design Pattern.